WeIR 물류 이송 로봇

WeGo Korea

목차

1. **WeIR**
2. **System Requirement**
3. **Start Up WeIR**
4. **WeIR Demo**
5. **Setting WeIR’s Network**
6. **WeIR Simulator**

01

WeIR

01 WeIR

Hardware List:

- Platform: Scout mini
- Controller: Jetson Xavier AGX Dev Kit
- Camera: Intel RealSense D435
- LiDAR: RPLiDAR A2 2ea

02

System Requirement

02 System Requirement

- Host Platform:
  - **Ubuntu Linux v18.04**
  - Note that a valid Internet connection and at least 10GB of disk space is needed for the complete installation of JetPack.
- Target Platform:
  - Jetson TX2
- Additional target requirements:
  - USB Micro-B cable
  - Ethernet Cable
  - Mouse, keyboard, HDMI

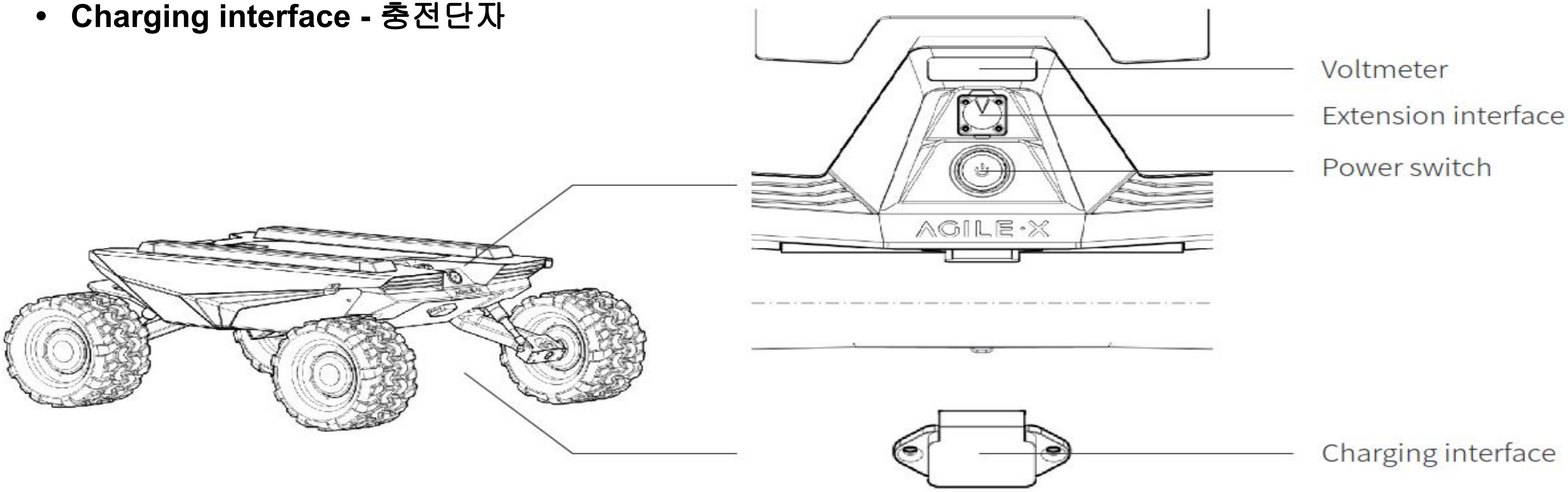
02 System Requirement

- JetPack 4.3
- ROS Operating System (Melodic)
- OpenCV 4
- WeIR Packages

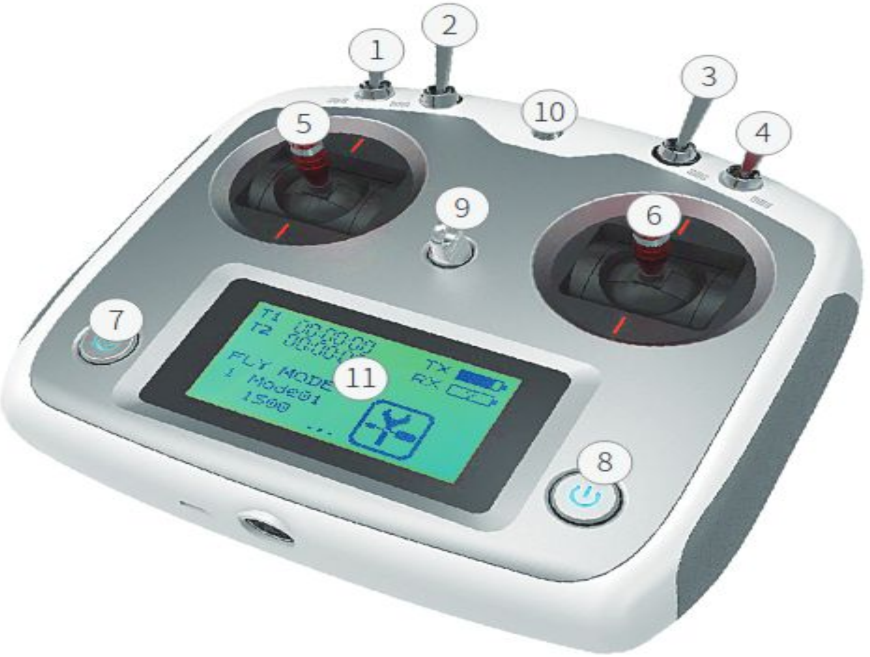
03

Startup WeIR

- **Scout mini**
- **Voltmeter - 현재 배터리 상태 및 잔량 확인**
- **Extension interface - CAN or 24V 전원 사용 가능**
- **Power switch - Scout mini 전원**
- **Charging interface - 충전단자**



- **Scout mini Controller**
- **전원 On / Off를 위해 7, 8 버튼을 길게 입력(1, 2, 3, 4는 위 올린 상태로 구동)**
- **수동 모드 조작을 위해 2를 중앙으로 이동한 후, 5를 이용하여 전, 후방 이동, 6을 이용하여 좌우 회전 조작 가능**
- **Serial or CAN을 이용한 주행을 위해서는 2를 위(CAN) 또는 아래(Serial)로 이동하면 자율 주행 모드 실행 가능**
- **3을 이용하여, 수동 모드 시, Scout mini의 조명을 변경 가능**
- **4를 이용하여, Scout mini의 최대 이동 속도 변경 가능(Speed Mode, Normal Mode)**



1. Lever SWA
2. Lever SWB
3. Lever SWC
4. Lever SWD
5. Left rocker
6. Right rocker
7. Power switch key 1
8. Power switch key 2
9. Mobile/Tablet fixing support interface
10. Ring interface
11. LCD panel

*When the user gets the RC transmitter, the settings have been available without having to be set separately.

ROS RPLidar package

- **rplidarNode - Get LaserScan Data from RPLidar**
  - **published topics**
    - scan (sensor_msgs/LaserScan) - LaserScan data from RPLidar
  - **subscribed topics**
    - None
  - **Parameters (Default)**
    - serial_port(string) - Serial port to Lidar (“/dev/ttyUSB0”)
    - serial_baudrate(int) - Serial Baudrate (115200 for A1, A2, 256000 for A3)
    - frame_id(string) - frame id for LaserScan (“laser”)
    - inverted(bool) - Lidar is mounted inverted or not (“false”)
    - angle_compensate(bool) - whether to compensate or not
    -

roslaunch rplidar_ros rplidar.launch

http://wiki.ros.org/rplidar

ROS Scout mini package

- **scout_base_node - Control Scout mini using CAN or Serial**
  - **published topics**
    - odom (nav_msgs/Odometry) - Wheel Odometry Data of Scout mini
    - scout_status (scout_msgs/ScoutStatus) - Status of Scout mini (Battery Voltage, Motor)
    - tf (tf2_msgs/TFMessage) - Transformation between odom to base_link
  - **subscribed topics**
    - cmd_vel (geometry_msgs/Twist) - Control Scout mini Linear and Angular Velocity
    - scout_light_control (scout_msgs/ScoutLightCmd) - Control Scout mini's Light
  - **Parameters (Default)**
    - ~port_name(string) - Serial or CAN port to Lidar ("can0")
    - ~base_frame(string) - Base frame id of Robot ("base_link")
    - ~odom_frame(string) - Odometry frame id of Robot ("odom")
    - ~simulated_robot(bool) - whether robot is simulated or real ("false")

roslaunch scout_bringup scout_minimal.launch

ROS RealSense package

- **realsense-ros package - ROS nodes for using Intel RealSense Cameras**
- **Published Topics**
  - /camera/color/camera_info
  - /camera/color/image_raw
  - /camera/depth/camera_info
  - /camera/depth/image_rect_raw
  - /camera/extrinsics/depth_to_color
  - /camera/extrinsics/depth_to_infra1
  - /camera/extrinsics/depth_to_infra2
  - /camera/infra1/camera_info
  - /camera/infra1/image_rect_raw
  - /camera/infra2/camera_info
  - /camera/infra2/image_rect_raw
  - /diagnostics

ROS RealSense package

- **realsense-ros package - ROS nodes for using Intel RealSense Cameras**
- **Parameters**
  - serial_no (특정 serial number의 장치에 접근, 없을 경우 인식된 Realsense 중 랜덤으로 접근)
  - usb_port_id (특정 USB port에 접근, 없을 경우, 장치 선택 시 사용하지 않음)
  - device_type (device_type이 포함된 장치에 접근, 없을 경우 고려하지 않음)
  - rosbag_filename (특정 rosbag file 로 부터 topic을 publish할 때 사용)
  - initial_reset (장치가 정상적으로 종료되지 않았거나, 펌웨어 관련 문제로 초기화가 필요한 경우, true로 설정 시, 사용 전에 reset을 진행)
  - align_depth (true로 설정 시, depth image와 align된 모든 이미지를 추가 topic으로 출력)
  - filters (아래의 옵션들 중 사용 가능, 콤마를 기준으로 구분)
  - colorizer - depth image를 16bit 대신 RGB 이미지로 출력
  - pointcloud - /camera/depth/color/points의 pointcloud topic을 추가로 출력, pointcloud의 texture는 rqt_reconfigure를 통해 수정하거나, pointcloud_texture_stream, pointcloud_texture_index를 통해 수정 가능

- https://github.com/IntelRealSense/realsense-ros
- **roslaunch realsense2_camera rs_camera.launch**

ROS Laser Filter package

- **scan_to_scan_filter_chain - Filter the LaserScan Data to LaserScan Data**
- **Published Topics**
  - /scan_filtered (sensor_msgs/LaserScan) - Filtered Data
- **Subscribed Topics**
  - /scan (sensor_msgs/LaserScan) - Original LaserScan Data

- **Parameters**
  - ~scan_filter_chain (list) - Filtering parameters

- **http://wiki.ros.org/laser_filters**
- **roslaunch laser_filters my_laser_filter.launch**

ROS Ira Laser Tools package

- **laser_scan_multi_merger - merge the many LaserScan Data to One LaserScan Data**
- **Published Topics**
  - /scan_multi (sensor_msgs/LaserScan) - Merged Data
  - /merged_cloud (sensor_msgs/PointCloud2) - Merged PointCloud
- **Subscribed Topics**
  - /scandx (sensor_msgs/LaserScan) - Original LaserScan Data
  - /scansx (sensor_msgs/LaserScan) - Original LaserScan Data
- **Parameters(Default)**
  - ~angle_increment (0.0058)
  - ~angle_max (2.0)
  - ~angle_min (-2.0)
  - ~range_min (0.3)
  - ~range_max (50.0)
  - ~scan_time (0.0333333)
  - ~time_increment (0.0058)
  - ~cloud_destination_topic (“/merged_cloud”)
  - ~destination_frame (“cart_fram”)
  - ~laserscan_topics (“/scandx /scansx”)
  - ~scan_destination_topic (“/scan_multi”)
- **https://github.com/iralabdisco/ira_laser_tools**
- **roslaunch ira_laser_tools laserscan_multi_merger.launch**

ROS Hector SLAM package

- **hector mapping - Lidar Only SLAM using LaserScan Data**
- **Published Topics**
    - /map_metadata (nav_msgs/MapMetaData) - Map Meta Data Type Map
    - /map (nav_msgs/OccupancyGrid) - OccupancyGrid Map Data
    - /slam_out_pose (geometry_msgs/PoseStamped) - Estimated Robot Pose without Covariance
    - /poseupdate (geometry_msgs/PoseWithCovarianceStamped) - Estimated Robot Pose with Gaussian Covariance
- **Subscribed Topics**
    - /scan (sensor_msgs/LaserScan) - Lidar LaserScan Data
    - /syscommand (std_msgs/String) - When Publish Message “reset”, Map and Robot Pose are reset

ROS Hector SLAM package

- **hector mapping - Lidar Only SLAM using LaserScan Data**
- **Parameters(Default)**
    - ~base_frame (base_link) - The name of the base frame of the robot
    - ~map_frame (map_link) - The name of map frame
    - ~odom_frame (odom) - The name of odom frame
    - ~map_resolution (0.025) - The map resolution (meter), the length of one cell
    - ~map_size (1024) - The Size of Map (1024 * 1024 Cells)
    - ~map_start_x (0.5) - Location of the Origin of Map X(0.0 ~ 1.0)
    - ~map_start_y (0.5) - Location of the Origin of Map Y(0.0 ~ 1.0)
    - ~map_update_distance_thresh (0.4) - Threshold for performing map updates (meter)
    - ~map_update_angle_thresh (0.9) - Threshold for performing map updates (radian)
    - ~map_pub_period (2.0) - The map publish period (second)
    - ~map_multi_res_levels (3) - The number of map multi-resolution grid levels
    -

ROS Hector SLAM package

- **hector mapping - Lidar Only SLAM using LaserScan Data**
- **Parameters(Default)**
  - ~update_factor_free (0.4) - updates of free cells in the range [0.0, 1.0]. A value of 0.5 means no change.
  - ~update_factor_occupied (0.9) - updates of occupied cells in the range [0.0, 1.0]. A value of 0.5 means no change.
  - ~laser_min_dist (0.4) - The minimum distance [m] for laser scan endpoints to be used by the system.
  - ~laser_max_dist (30.0) - The maximum distance [m] for laser scan endpoints to be used by the system.
  - ~laser_z_min_value (-1.0) - The minimum height [m] relative to the laser scanner frame for laser scan endpoints to be used by the system. Scan endpoints lower than this value are ignored.
  - ~laser_z_max_value (1.0) - The maximum height [m] relative to the laser scanner frame for laser scan endpoints to be used by the system. Scan endpoints higher than this value are ignored.
  - ~pub_map_odom_transform (true) - Determine if the map->odom transform should be published by the system.
  - ~output_timing (false) - Output timing information for processing of every laser scan via ROS_INFO.
  - ~scan_subscriber_queue_size (5) - The queue size of the scan subscriber. This should be set to high values (for example 50) if log-files are played back to hector_mapping at faster than realtime speeds.
  - ~pub_map_scanmatch_transform (true) - Determines if the scanmatcher to map transform should be published to tf. The frame name is determined by the 'tf_map_scanmatch_transform_frame_name' parameter.
  - ~tf_map_scanmatch_transform_frame_name (scanmatcher_frame) - The frame name when publishing the scanmatcher to map transform as described in the preceding parameter.
- **http://wiki.ros.org/hector_mapping**
- **roslaunch hector_slam_launch tutorial.launch**

ROS Map Server package

- **map_server - Start Server which provide topics and services about Map**
- **Published Topics**
  - map_metadata (nav_msgs/MapMetaData) - MapMetaData Type Map
  - map (nav_msgs/OccupancyGrid) - OccupancyGrid Type Map
- **Subscribed Topics**
  - None
- **Parameters(Default)**
  - ~frame_id (“map”)
- **http://wiki.ros.org/map_server**
- **rosrun map_server map_server <map_path>**

ROS Map Server package

- **map_saver - Save map topic to file**
- **Published Topics**
  - None
- **Subscribed Topics**
  - map (nav_msgs/OccupancyGrid) - OccupancyGrid Type Map
- **Parameters(Default)**
  - None
- **http://wiki.ros.org/map_server**
- **rosrun map_server map_saver -f <map_name>**

ROS AMCL package

- **amcl - adaptive monte carlo localization**
- **Published Topics**
  - amcl_pose (geometry_msgs/PoseWithCovarianceStamped) - Robot's estimated pose in the map, with covariance.
  - particlecloud (geometry_msgs/PoseArray) - The set of pose estimates being maintained by the filter.
  - tf (tf/tfMessage) - Publishes the transform from odom (~odom_frame_id parameter) to map.
- **Subscribed Topics**
  - scan (sensor_msgs/LaserScan) - Lidar Laser scan data.
  - tf (tf/tfMessage) - Transforms.
  - initialpose (geometry_msgs/PoseWithCovarianceStamped) - Mean and covariance with which to (re-)initialize the PF.
  - map (nav_msgs/OccupancyGrid) - When the use_map_topic parameter is set, AMCL subscribes to map topic to retrieve the map used for laser-based localization

ROS AMCL package

- **amcl - adaptive monte carlo localization**
- **Parameters(Default) -** Overall filter parameters
    - ~min_particles (100) - Minimum allowed number of particles.
    - ~max_particles (5000) - Maximum allowed number of particles.
    - ~kld_err (0.01) - Maximum error between the true distribution and the estimated distribution.
    - ~kld_z (0.99) - Upper standard normal quantile for (1 - p), where p is the probability that the error on the estimated distrubition will be less than kld_err.
    - ~update_min_d (0.2 meters) - Translational movement required before performing a filter update.
    - ~update_min_a (π/6.0 radians) - Rotational movement required before performing a filter update.
    - ~resample_interval (2) - Number of filter updates required before resampling.
    - ~transform_tolerance (0.1 seconds) - Time with which to post-date the transform that is published, to indicate that this transform is valid into the future.
    - ~recovery_alpha_slow (0.0 (disabled)) - Exponential decay rate for the slow average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.001.
    - ~recovery_alpha_fast (0.0 (disabled)) - Exponential decay rate for the fast average weight filter, used in deciding when to recover by adding random poses. A good value might be 0.1.
    - ~initial_pose_x (0.0 meters) - Initial pose mean (x), used to initialize filter with Gaussian distribution.
    - ~initial_pose_y (0.0 meters) - Initial pose mean (y), used to initialize filter with Gaussian distribution.
    - ~initial_pose_a (0.0 radians) - Initial pose mean (yaw), used to initialize filter with Gaussian distribution.

ROS AMCL package

- **amcl - adaptive monte carlo localization**
- **Parameters(Default) -** Overall filter parameters
  - ~initial_cov_xx (0.5*0.5 meters) - Initial pose covariance (x*x), used to initialize filter with Gaussian distribution.
  - ~initial_cov_yy (0.5*0.5 meters) - Initial pose covariance (y*y), used to initialize filter with Gaussian distribution.
  - ~initial_cov_aa ((π/12)*(π/12) radian) - Initial pose covariance (yaw*yaw), used to initialize filter with Gaussian distribution.
  - ~gui_publish_rate (double, default: -1.0 Hz) - Maximum rate (Hz) at which scans and paths are published for visualization, -1.0 to disable.
  - ~save_pose_rate (double, default: 0.5 Hz) - Maximum rate (Hz) at which to store the last estimated pose and covariance to the parameter server, in the variables ~initial_pose_* and ~initial_cov_*. This saved pose will be used on subsequent runs to initialize the filter. -1.0 to disable.
  - ~use_map_topic (bool, default: false) - When set to true, AMCL will subscribe to the map topic rather than making a service call to receive its map. **New in navigation 1.4.2**
  - ~first_map_only (bool, default: false) - When set to true, AMCL will only use the first map it subscribes to, rather than updating each time a new one is received. **New in navigation 1.4.2**
  - ~selective_resampling (bool, default: false) - When set to true, will reduce the resampling rate when not needed and help avoid particle deprivation. The resampling will only happen if the effective number of particles (*N_eff = 1/(sum(k_i^2))*) is lower than half the current number of particles. Reference: *Grisetti, Giorgio, Cyrill Stachniss, and Wolfram Burgard. "Improved techniques for grid mapping with rao-blackwellized particle filters." IEEE transactions on Robotics 23.1 (2007): 34.*

ROS AMCL package

- **amcl - adaptive monte carlo localization**
- **Parameters(Default) -** Laser model parameters
  - ~laser_min_range (-1.0) - Minimum scan range to be considered, -1.0 will cause the laser's reported minimum range to be used.
  - ~laser_max_range (-1.0) - Maximum scan range to be considered; -1.0 will cause the laser's reported maximum range to be used.
  - ~laser_max_beams (30) - How many evenly-spaced beams in each scan to be used when updating the filter.
  - ~laser_z_hit (0.95) - Mixture weight for the z_hit part of the model.
  - ~laser_z_short (0.1) - Mixture weight for the z_short part of the model.
  - ~laser_z_max (0.05) - Mixture weight for the z_max part of the model.
  - ~laser_z_rand (0.05) - Mixture weight for the z_rand part of the model.
  - ~laser_sigma_hit (0.2 meters) - Standard deviation for Gaussian model used in z_hit part of the model.
  - ~laser_lambda_short (0.1) - Exponential decay parameter for z_short part of model.
  - ~laser_likelihood_max_dist (2.0 meters) - Maximum distance to do obstacle inflation on map, for use in likelihood_field model.
  - ~laser_model_type ("likelihood_field") - Which model to use, either beam, likelihood_field, or likelihood_field_prob (same as likelihood_field but incorporates the beamskip feature, if enabled).

ROS AMCL package

- **amcl - adaptive monte carlo localization**
- **Parameters(Default) -** Odometry model parameters
  - ~odom_model_type ("diff") - Which model to use, either "diff", "omni", "diff-corrected" or "omni-corrected".
  - ~odom_alpha1 (0.2) - Specifies the expected noise in odometry's rotation estimate from rotational component of robot's motion.
  - ~odom_alpha2 (0.2) - Specifies the expected noise in odometry's rotation estimate from translational component of robot's motion.
  - ~odom_alpha3 (0.2) - Specifies the expected noise in odometry's translation estimate from translational component of robot's motion.
  - ~odom_alpha4 (0.2) - Specifies the expected noise in odometry's translation estimate from the rotational component of robot's motion.
  - ~odom_alpha5 (0.2) - Translation-related noise parameter (only used if model is "omni").
  - ~odom_frame_id ("odom") - Which frame to use for odometry.
  - ~base_frame_id ("base_link") - Which frame to use for the robot base
  - ~global_frame_id ("map") - The name of the coordinate frame published by the localization system
  - ~tf_broadcast (true) - Set this to false to prevent amcl from publishing the transform between the global frame and the odometry frame.

- http://wiki.ros.org/amcl
- roslaunch scout_mini_2dnav amcl.launch

ROS Move Base package

- **move_base - planning & control the robot**
- **Published Topics**
    - cmd_vel (geometry_msgs/Twist) - A stream of velocity commands meant for execution by a mobile base.
- **Subscribed Topics**
    - move_base_simple/goal (geometry_msgs/PoseStamped) - Provides a non-action interface to move_base for users that don't care about tracking the execution status of their goals.
- **Parameters(Default)**
    - ~base_global_planner ("navfn/NavfnROS") - The name of the plugin for the global planner to use with move_base, see pluginlib documentation for more details on plugins. This plugin must adhere to the nav_core::BaseGlobalPlanner interface specified in the nav_core package.
    - ~base_local_planner ("base_local_planner/TrajectoryPlannerROS") - The name of the plugin for the local planner to use with move_base see pluginlib documentation for more details on plugins. This plugin must adhere to the nav_core::BaseLocalPlanner interface specified in the nav_core package.
    - ~recovery_behaviors ([{name: conservative_reset, type: clear_costmap_recovery/ClearCostmapRecovery}, {name: rotate_recovery, type: rotate_recovery/RotateRecovery}, {name: aggressive_reset, type: clear_costmap_recovery/ClearCostmapRecovery}] - A list of recovery behavior plugins to use with move_base, see pluginlib documentation for more details on plugins. These behaviors will be run when move_base fails to find a valid plan in the order that they are specified. After each behavior completes, move_base will attempt to make a plan. If planning is successful, move_base will continue normal operation. Otherwise, the next recovery behavior in the list will be executed. These plugins must adhere to the nav_core::RecoveryBehavior interface specified in the nav_core package.

ROS Move Base package

- **move_base - planning & control the robot**
- **Parameters(Default)**
  - ~controller_frequency (20.0) - The rate in Hz at which to run the control loop and send velocity commands to the base.
  - ~planner_patience (5.0) - How long the planner will wait in seconds in an attempt to find a valid plan before space-clearing.
  - ~controller_patience (15.0) - How long the controller will wait in seconds without receiving a valid control before space-clearing.
  - ~conservative_reset_dist (3.0) - The distance away from the robot in meters beyond which obstacles will be cleared from the costmap when attempting to clear space in the map. Note, this parameter is only used when the default recovery behaviors are used for move_base
  - ~recovery_behavior_enabled (true) - Whether or not to enable the move_base recovery behaviors to attempt to clear out space.
  - ~clearing_rotation_allowed (true) - Determines whether or not the robot will attempt an in-place rotation when attempting to clear space.
  - ~shutdown_costmaps (false) - Determines whether or not to shutdown the costmaps of the node when move_base is in an inactive state
  - ~oscillation_timeout (0.0) - How long in seconds to allow for oscillation before executing recovery behaviors. A value of 0.0 corresponds to an infinite timeout.
  - ~oscillation_distance (0.5) - How far in meters the robot must move to be considered not to be oscillating. Moving this far resets the timer counting up to the ~oscillation_timeout
  - ~planner_frequency (0.0) - The rate in Hz at which to run the global planning loop. If the frequency is set to 0.0, the global planner will only run when a new goal is received or the local planner reports that its path is blocked
  - ~max_planning_retries (-1) - How many times to allow for planning retries before executing recovery behaviors. A value of -1.0 corresponds to an infinite retries.
- http://wiki.ros.org/move_base
- roslaunch scout_mini_2dnav move_base_only.launch

04

WeIR Demo

Start Robot using Serial

```
$ sudo chmod 666 /dev/tty*
$ roslaunch scout_bringup scout_minimal.launch → Scout mini 제어
$ roslaunch rplidar_ros rplidar_all.launch → RPLidar Running
$ roslaunch laser_filters my_laser_filters.launch → Filtering LaserScan
$ roslaunch ira_laser_tools laserscan_multi_merger.launch → Merging LaserScan Data
```

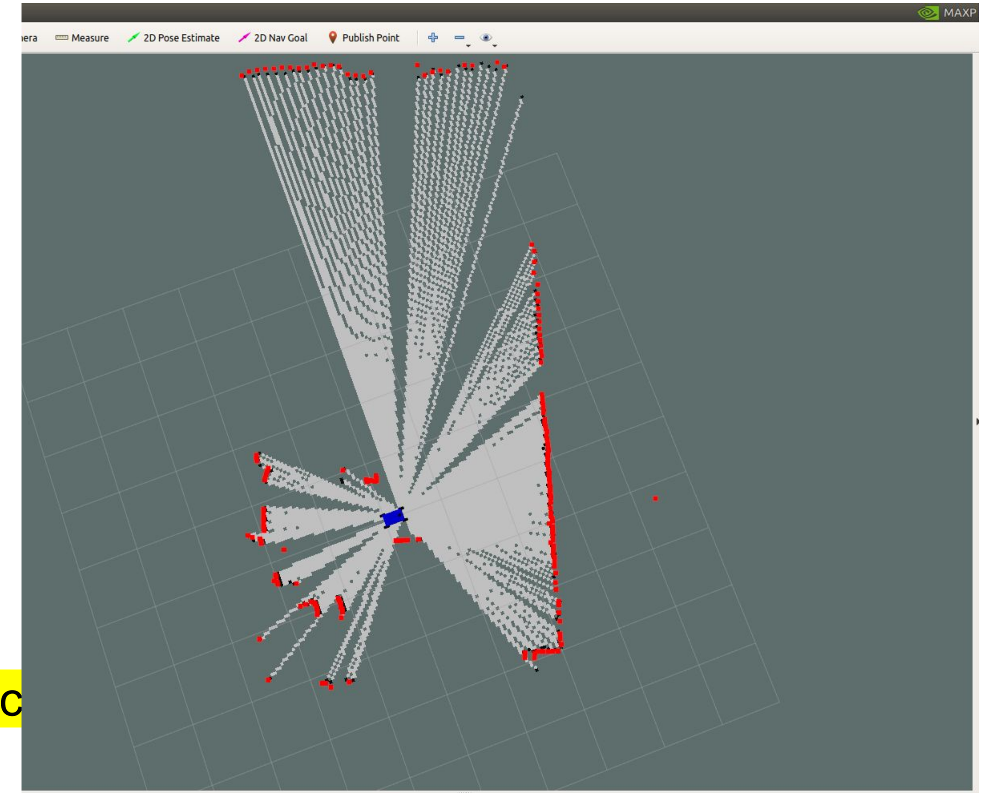
Start Robot using CAN

```
$ sudo chmod 666 /dev/tty*
$ rosrun scout_bringup bringup_can2usb.bash
$ roslaunch scout_bringup scout_minimal.launch → Scout mini 제어
$ roslaunch rplidar_ros rplidar_all.launch → RPLidar Running
$ roslaunch laser_filters my_laser_filters.launch → Filtering LaserScan
$ roslaunch ira_laser_tools laserscan_multi_merger.launch → Merging LaserScan Data
```

Mapping

After Start Robot (p. 30 ~ 31)

```
$ roslaunch wego mapping.launch
```



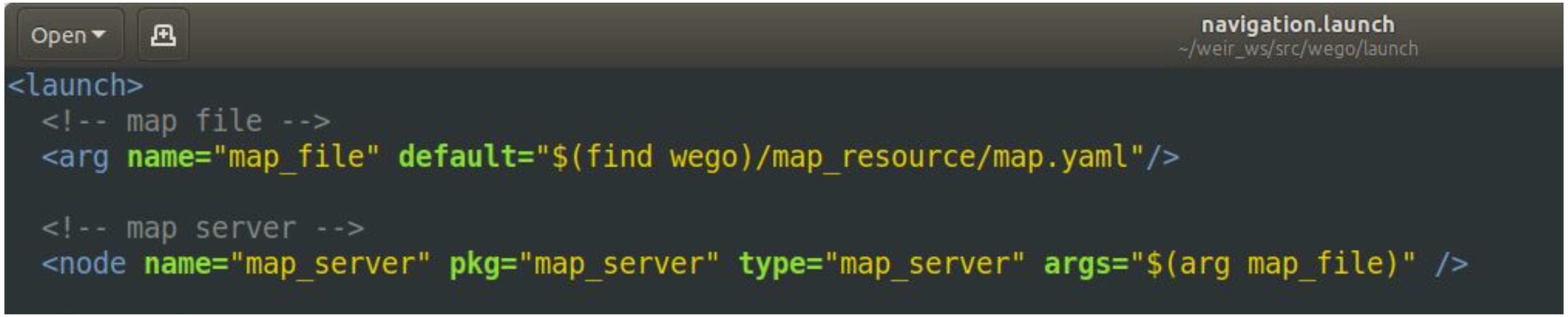
지도 저장

- Open New Terminal

```
$ rosrun map_server map_saver -f “MapName”
$ sudo mv MapName.* ~/weir_ws/src/wego/map_resourc
```

Localization & Navigation

- map_saver를 이용한 Map 저장 후, Mapping 및 모든 노드 종료
- Map 변경
  - ~/weir_ws/src/wego/launch/navigation.launch 파일을 수정
  - <arg name= “map_file” default= “$(find wego)/map_resource/<MapName.yaml>” />



```
<launch>
  <!-- map file -->
  <arg name="map_file" default="$(find wego)/map_resource/map.yaml"/>

  <!-- map server -->
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg map_file)" />
```

Localization & Navigation

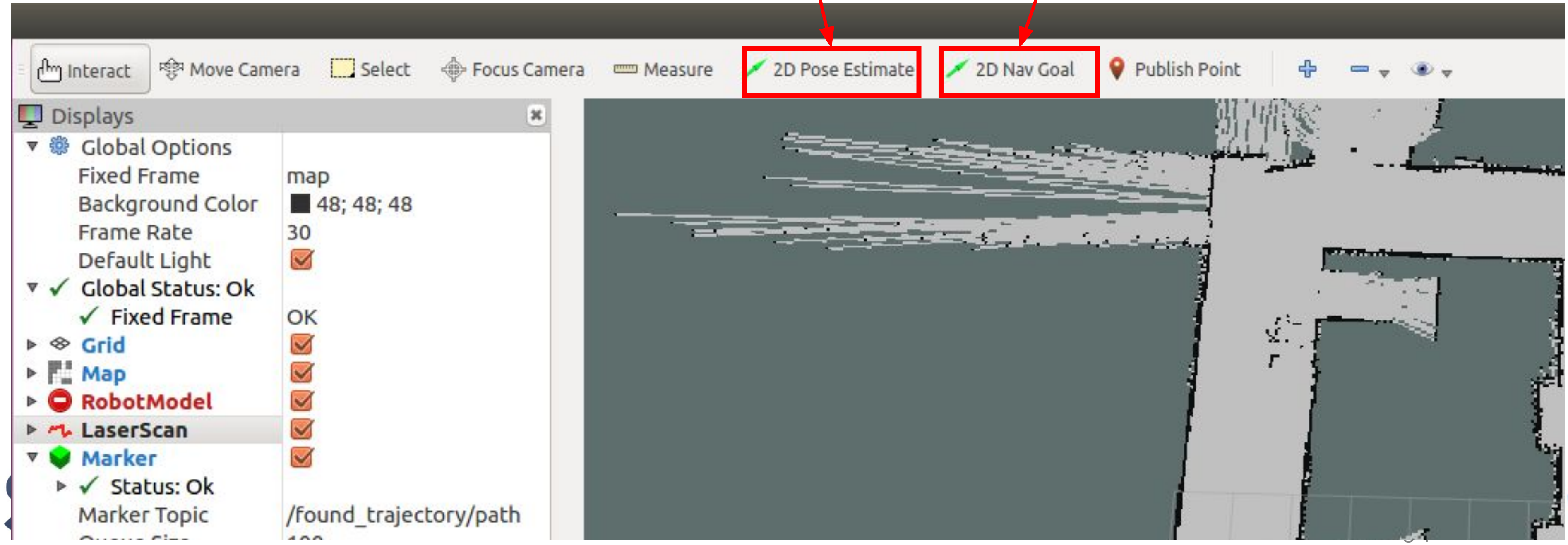
After Start Robot (p. 30 ~ 31)

```
$ roslaunch wego navigation.launch
```

- Rviz Setting
  - 2D Pose Estimate : 출발 방향 지정
  - 2D Nav Goal : 차량 주행 위치 지정

초기위치

최종위치



Displays

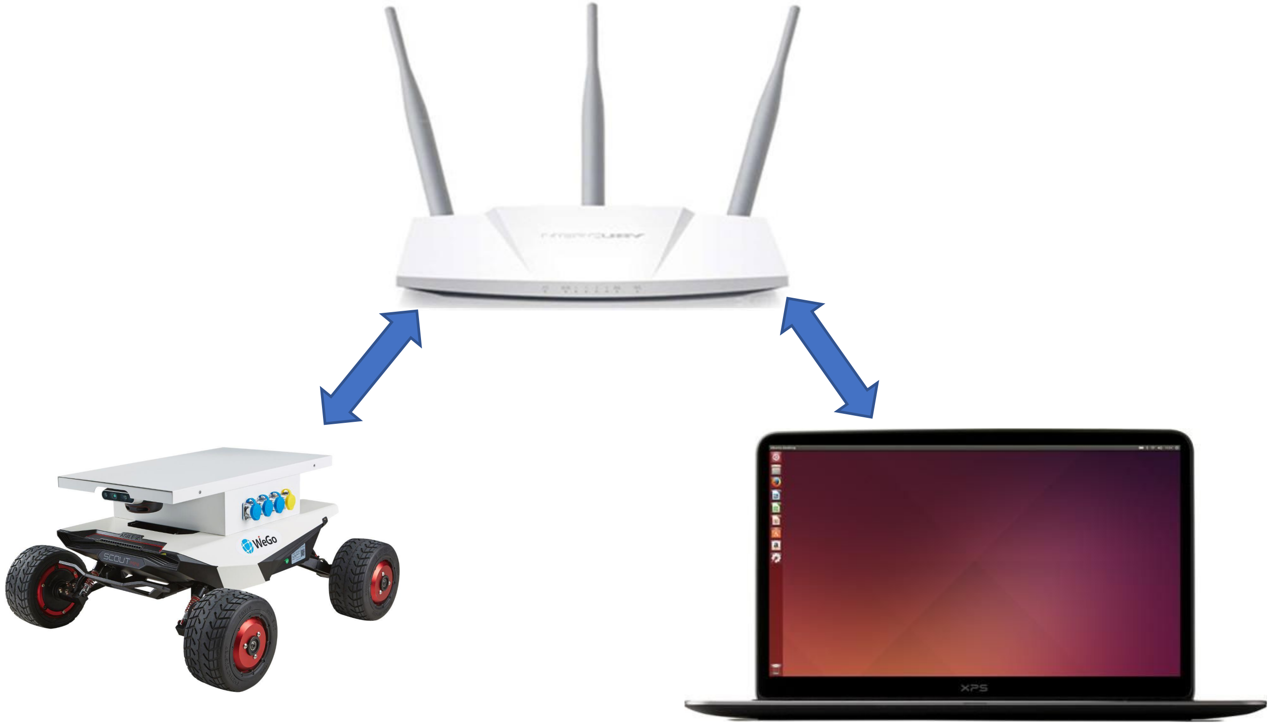
- Global Options
  - Fixed Frame: map
  - Background Color: 48; 48; 48
  - Frame Rate: 30
  - Default Light: ☑
- ✓ Global Status: Ok
  - ✓ Fixed Frame: OK
- Grid: ☑
- Map: ☑
- RobotModel: ☑
- LaserScan: ☑
- Marker: ☑
  - ✓ Status: Ok
  - Marker Topic: /found_trajectory/path

05

Setting WeIR’s Network

Setting up WeIR’s network

Remote PC : Ubuntu와 ROS가 설치된 상태에서만 진행가능



192.168.0.100

(WeIR’s IP)

192.168.0.71

(PC’s IP)

Setting up WeIR’s network

Remote PC :
동일한 네트워크에 WeIR와 PC 연결

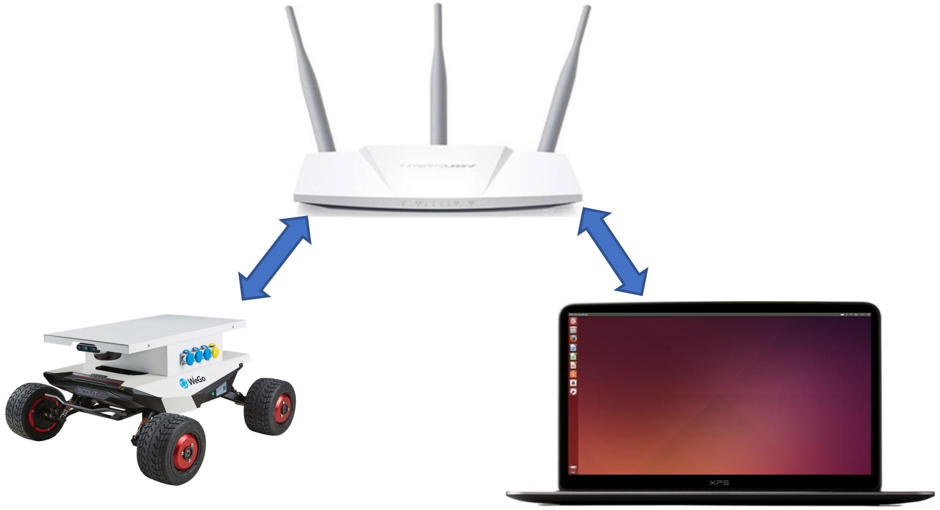
Remote Terminal

```
$ ssh nvidia@WeIR’s IP
$ nvidia passwd : nvidia
$ export ROS_MASTER_URI=http:// WeIR’s IP:11311
$ export ROS_IP=WeIR’s IP
$ roscore
```

Host Terminal

```
$ export ROS_MASTER_URI=http:// WeIR’s IP:11311
$ export ROS_IP=PC’s IP
$ rostopic list
```

연결 시, Topic 목록 확인 가능



06

WeIR Simulator

WeIR Simulator

- Unity 기반의 물리엔진이 적용된 Simulator
- ROS Bridge를 통해 연결하며, ROS 기반의 센서 데이터 취득 및 제어가 가능
- 현재 제작 진행 중